

# CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning

Hung Le\*, Yue Wang\*, Akhilesh Deepak Gotmare, Silvio Savarese, Steven C.H. Hoi

*Presenting at NeurIPS 2022*

*Thirty-sixth Conference on Neural Information Processing Systems, New Orleans*

Code and Model: <https://github.com/salesforce/CodeRL>

Paper: <https://arxiv.org/abs/2207.01780>

Blog: <https://blog.salesforceairesearch.com/coderl/>





CodeRL

# 1. CodeRL improves pretrained LMs for program synthesis by incorporating unit test signals in model training

LM is pretrained on massive public source code data

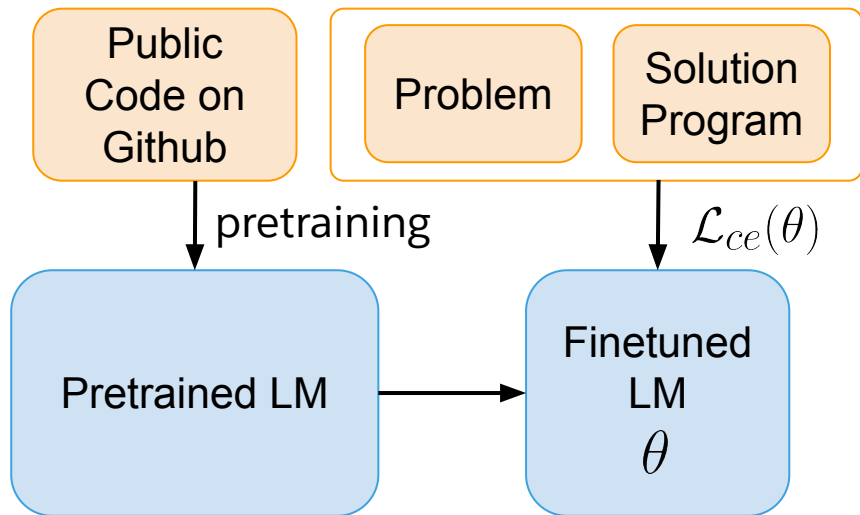
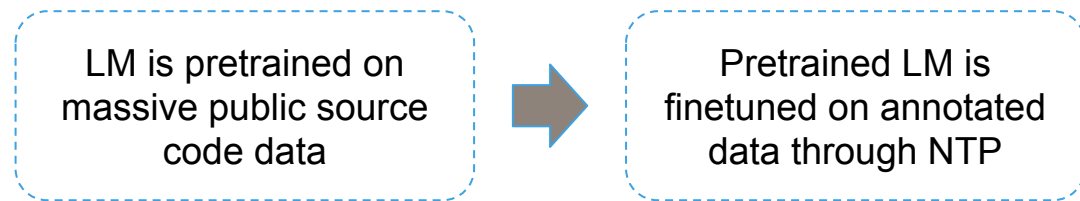
Public  
Code on  
Github

pretraining

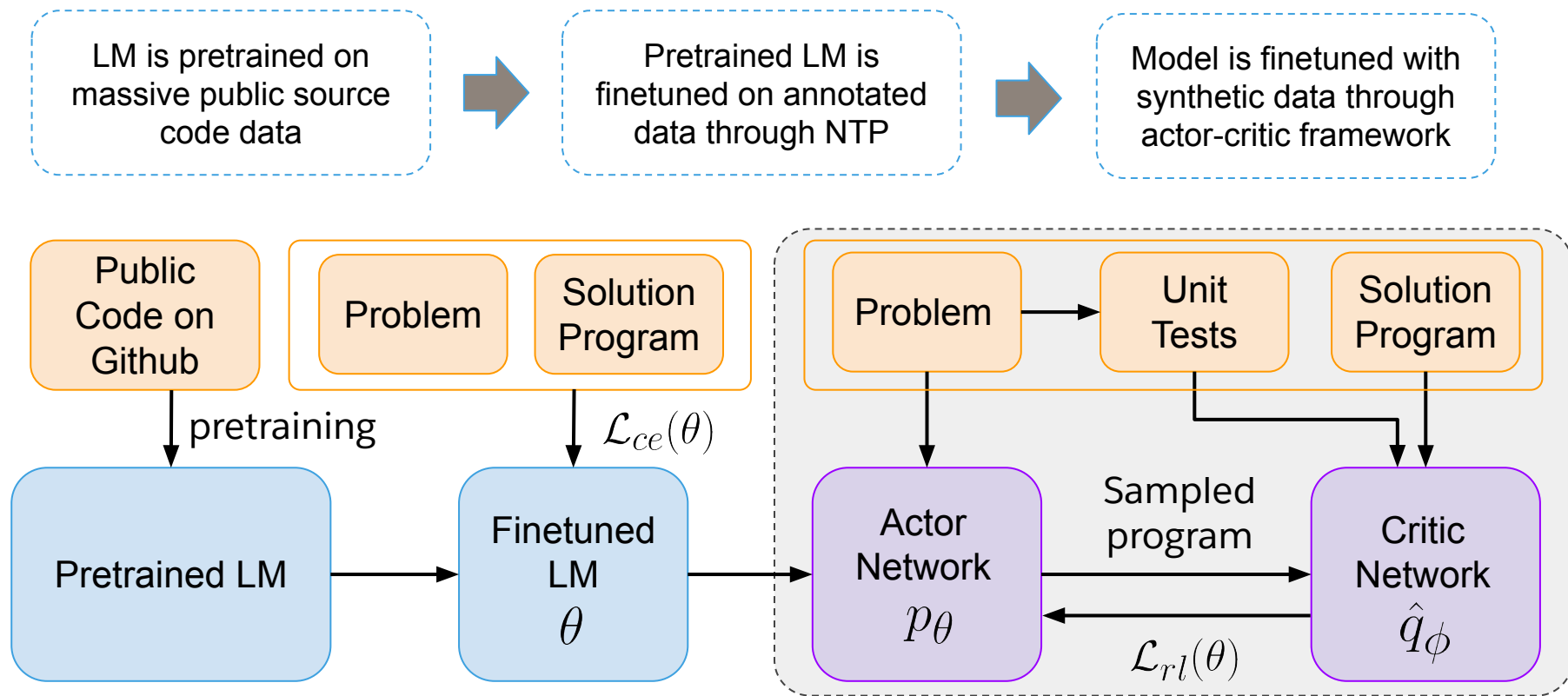
Pretrained LM

```
graph TD; A[Public Code on Github] -- pretraining --> B[Pretrained LM];
```

# 1. CodeRL improves pretrained LMs for program synthesis by incorporating unit test signals in model training

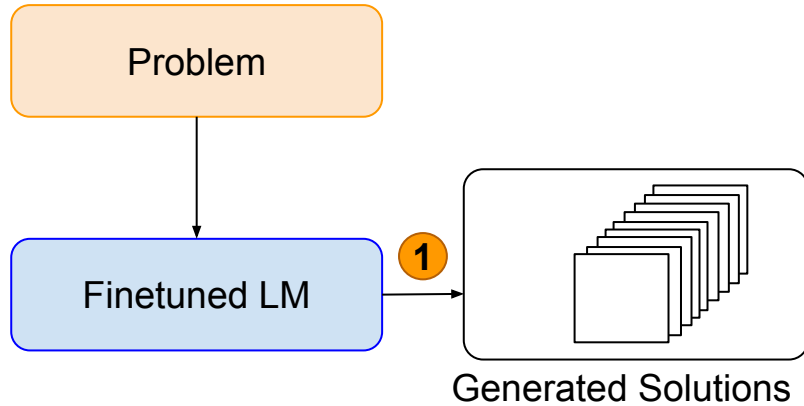


# 1. CodeRL improves pretrained LMs for program synthesis by incorporating unit test signals in model training

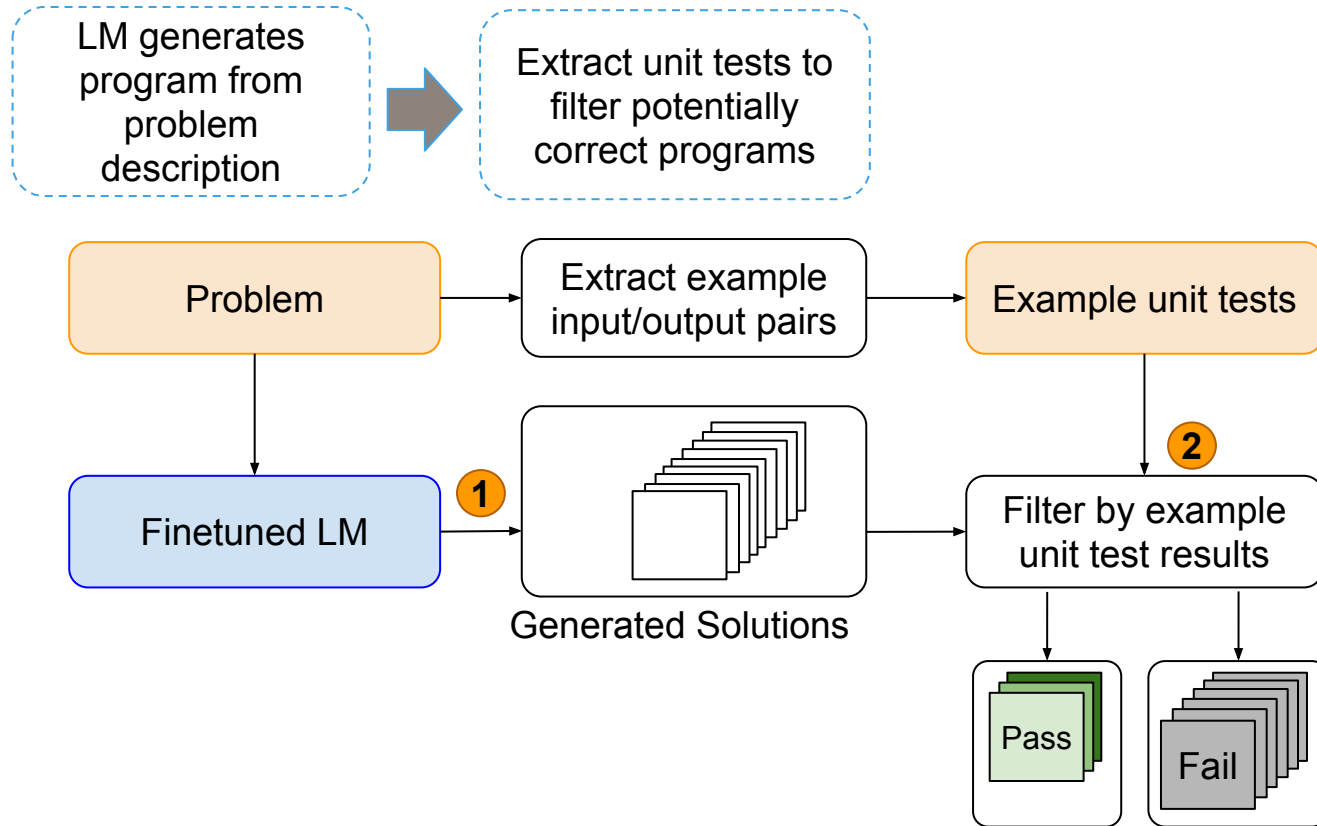


## 2. CodeRL improves program generation by re(generating) programs based on their initial example unit test results

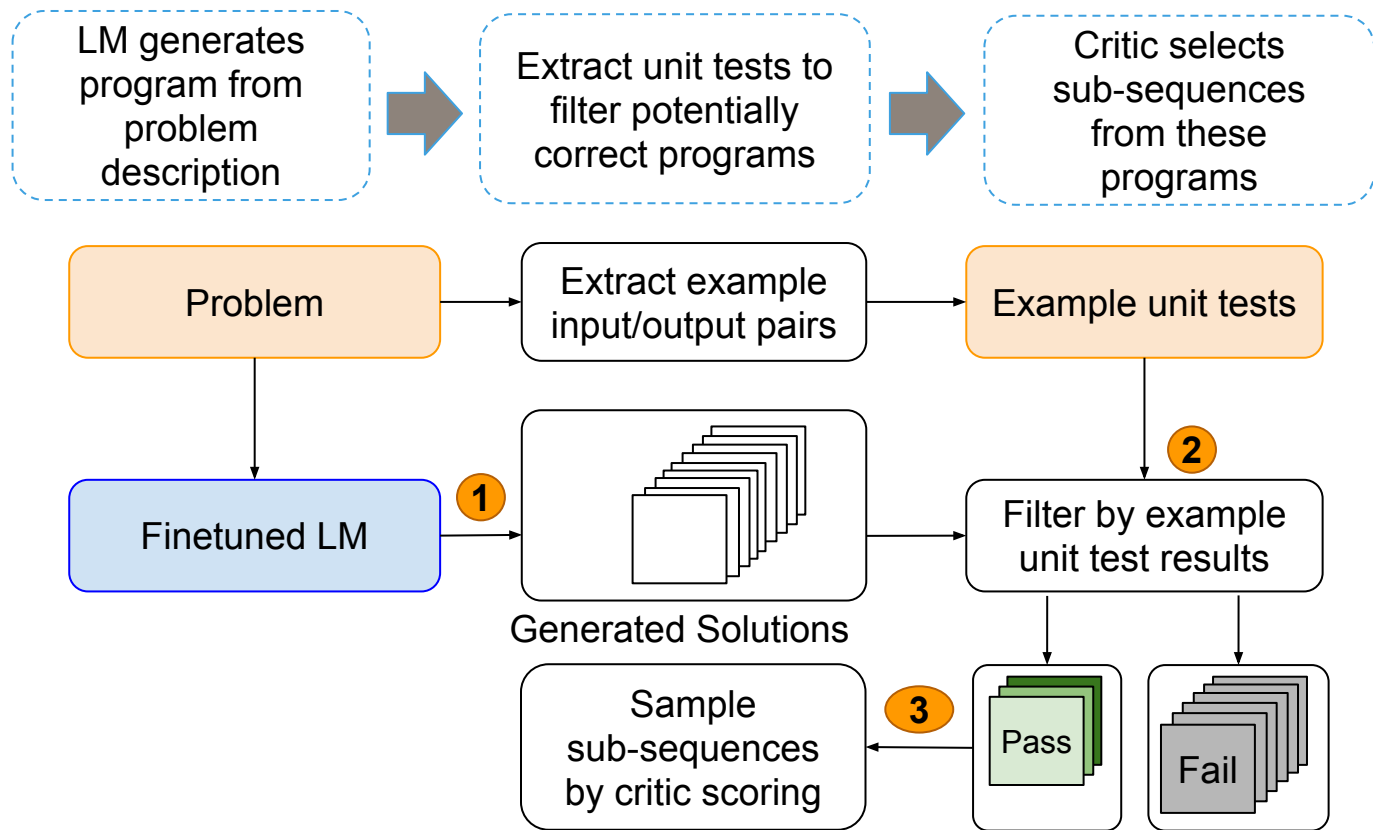
LM generates program from problem description



## 2. CodeRL improves program generation by re(generating) programs based on their initial example unit test results

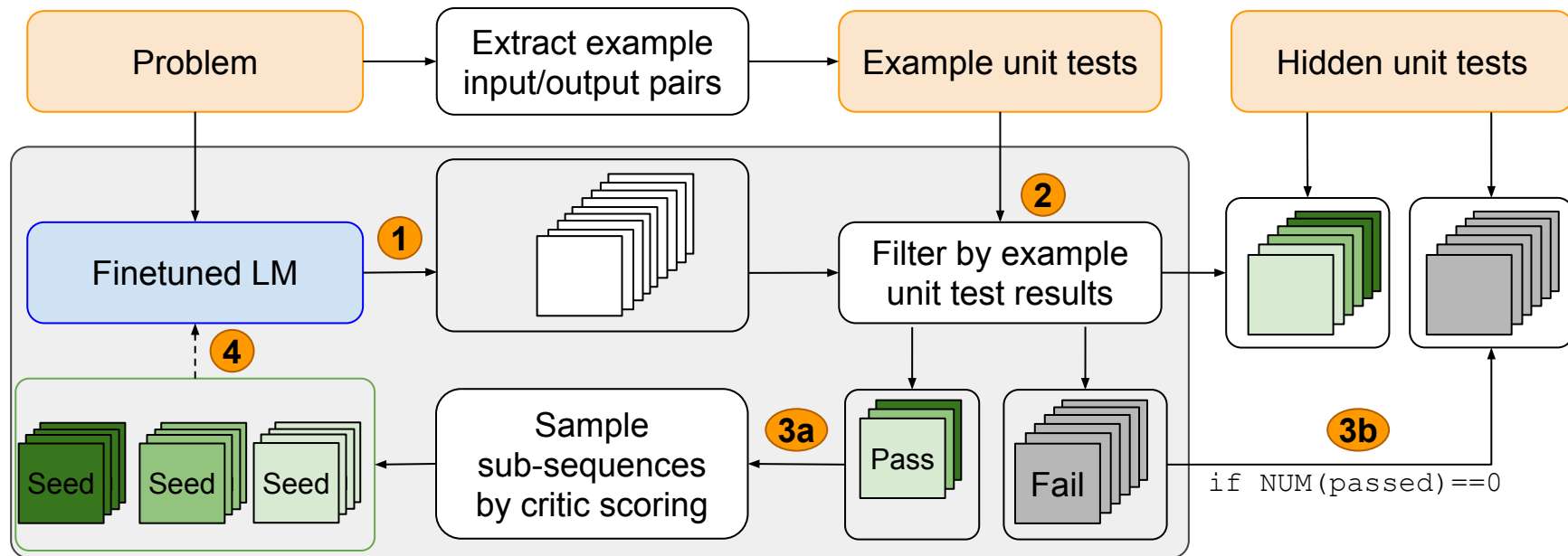
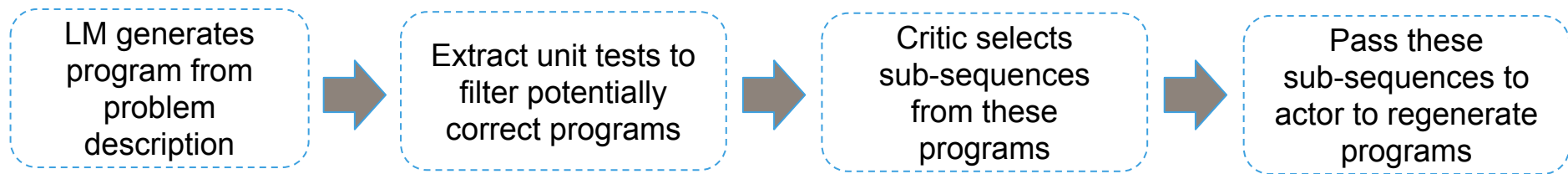


## 2. CodeRL improves program generation by re(generating) programs based on their initial example unit test results

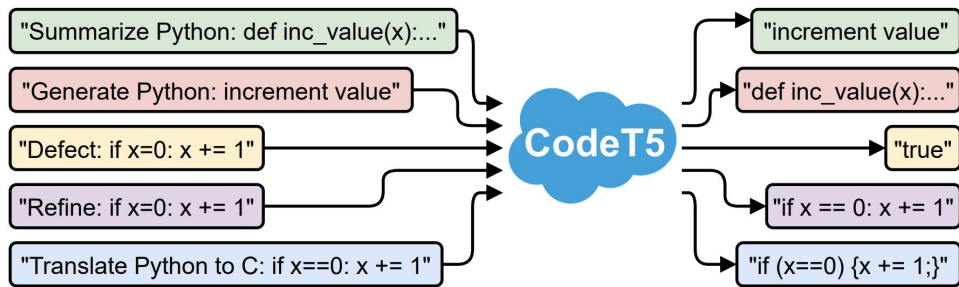




## 2. CodeRL improves program generation by re(generating) programs based on their initial example unit test results



### 3. We extend CodeT5 with larger pretraining data, better pretraining objective, and larger model size



**Pretraining Data**

GitHub Code dataset



**Pretraining Objective**

Next token prediction



**Model Size**

Up to 770M

# We evaluate CodeRL on two benchmarks: one with competitive programming tasks and another with beginner-level programming tasks

## Problem Specification

A string is a palindrome if it reads the same from the left to the right and from the right to the left....If there is such a substring in \$\$\$ that is not a palindrome, print the maximum length of such a substring....

**Example Input and Output:** Input: 'hannah' Output: 5

## Solution Program

```
1 s = input()
2 ans = 0
3 for i in range(len(s)):
4     for j in range(i + 1, len(s) + 1):
5         if s[i:j] != s[i:j][::-1]:
6             ans = max(ans, j - i)
7 print(ans)
```

## Unit Tests

Input: wuffuw  
Input: iiiiiii

Output: 5  
Output: 0...

APPS

Write a python function to check if a given number is one less than twice its reverse. Your code should satisfy these tests:

```
assert check(70) == False
assert check(23) == False
assert check(73) == True
```

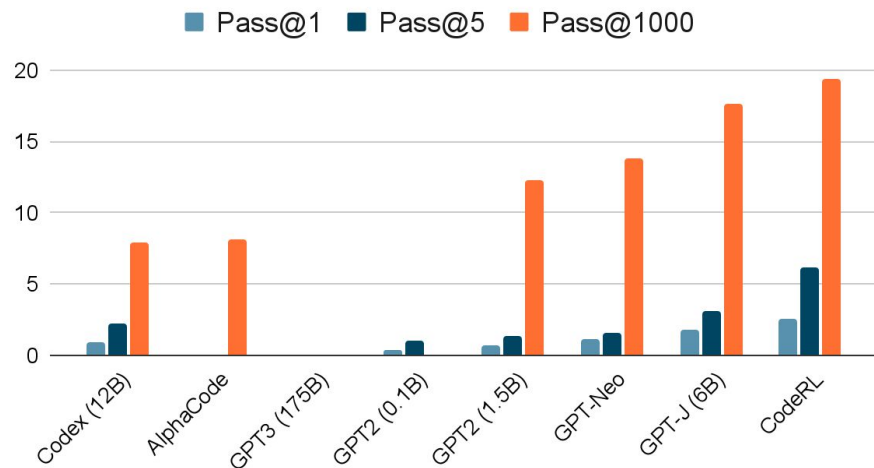
Write a Python function to sort the given array by using merge sort. Your code should satisfy these tests:

```
assert merge_sort([3, 4, 2, 6, 5, 7, 1, 9]) == [1, 2, 3, 4, 5, 6, 7, 9]
assert merge_sort([7, 25, 45, 78, 11, 33, 19]) == [7, 11, 19, 25, 33, 45, 78]
assert merge_sort([3, 1, 4, 9, 8]) == [1, 3, 4, 8, 9]
```

MBPP

# CodeRL achieves new SoTA performance in programming tasks in both APPS and MBPP (zero-shot)

Performance by pass@k on APPS

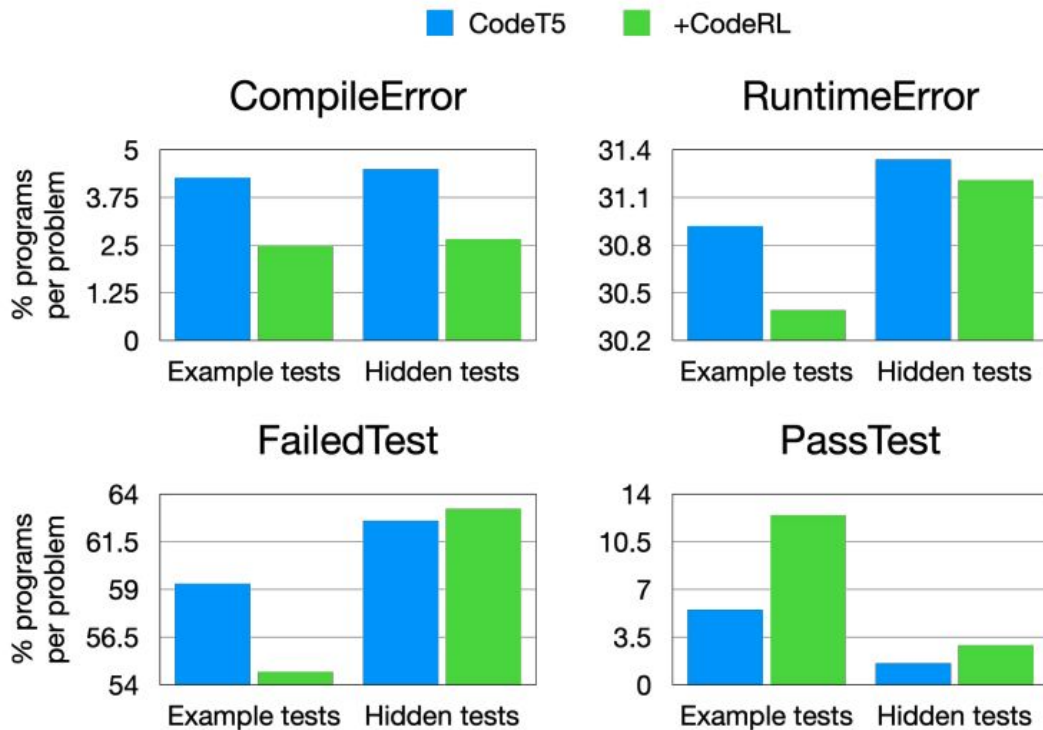


APPS

Model	Size	<i>pass@80</i>
GPT	224M	7.2
GPT	422M	12.6
GPT	1B	22.4
GPT	4B	33.0
GPT	8B	40.6
GPT	68B	53.6
GPT	137B	61.4
CodeRL+CodeT5 (ZS)	770M	<b>63.0</b>

MBPP

# CodeRL models show improvement over baseline models by generating more functionally correct programs



CodeRL models  
show improvement  
over baseline  
models by  
generating more  
functional correct  
programs

### Input Problem

Allen has a LOT of money. He has  $n$  dollars in the bank. For security reasons, he wants to withdraw it in cash (we will not disclose the reasons here). The denominations for dollar bills are \$1\$, \$5\$, \$10\$, \$20\$, \$100\$. What is the minimum number of bills Allen could receive after withdrawing his entire balance?

CodeRL models  
show improvement  
over baseline  
models by  
generating more  
functional correct  
programs

### Example Unit Tests

```
Input: 125
Output: 3

Input: 43
Output: 5

...
```

### Hidden Unit Tests

```
Input: 74
Output: 8

Input: 82655
Output: 830

...
```

### Ground-truth Program

```
1 n = int(input())
2
3 ans = 0
4 ans += n//100
5 n %= 100
6 ans += (n // 20)
7 n %= 20
8 ans += n //10
9 n %= 10
10 ans += n//5
11 n%= 5
12 ans += n//1
13
14 print(ans)
```

Example tests: **Passed**  
Hidden tests: **Passed**

### CodeT5 Output Program

```
1 n = int(input())
2
3 a = defaultdict(int)
4
5 for i in range(1, 101):
6     a[i] = n // i
7     n = n % i
8
9 ans = 0
10 for i in range(1, 101):
11     if n < a[i]:
12         ans += a[i]
13         n -= a[i]
14
15 print(ans)
```

Example tests: **Failed**  
Hidden tests: **Failed**

CodeRL models show improvement over baseline models by generating more functional correct programs

### CodeRL+CodeT5 Output Program (before CS)

```
1 def main():
2     n = int(input())
3     cnt = 0
4     while n > 0:
5         if n % 100 == 0:
6             n -= 100
7             cnt += 1
8         elif n % 20 == 0:
9             n -= 20
10            cnt += 1
11            elif n % 5 == 0:
12                n -= 5
13                cnt += 1
14            elif n % 10 == 0:
15                n -= 10
16                cnt += 1
17            elif n > 0:
18                n -= 1
19                cnt += 1
20        print(cnt)
21
22
23 def __starting_point():
24     main()
25
26 __starting_point()
```

Example tests: **Passed**  
Hidden tests: **Failed**

### CodeRL+CodeT5 Output Program (after CS)

```
1 def main():
2     n = int(input())
3     cnt = 0
4     while n > 0:
5         if n % 100 == 0:
6             n -= 100
7             cnt += 1
8         elif n % 20 == 0:
9             n -= 20
10            cnt += 1
11            elif n % 10 == 0:
12                n -= 10
13                cnt += 1
14            elif n % 5 == 0:
15                n -= 5
16                cnt += 1
17            else:
18                n -= 1
19                cnt += 1
20        print(cnt)
21
22
23 def __starting_point():
24     main()
25
26 __starting_point()
```

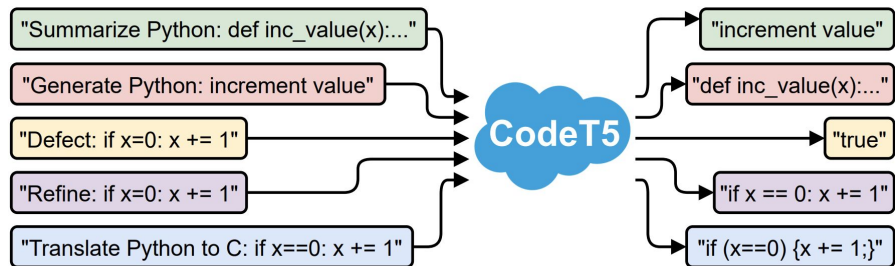
Example tests: **Passed**  
Hidden tests: **Passed**



In summary, CodeRL is a general framework that integrates pretrained LMs and RL holistically for program synthesis



RL framework



Pretrained LMs such as CodeT5

# CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning

Hung Le\*, Yue Wang\*, Akhilesh Deepak Gotmare, Silvio Savarese, Steven C.H. Hoi

***THANK YOU!***

Code and Model: <https://github.com/salesforce/CodeRL>

Paper: <https://arxiv.org/abs/2207.01780>

Blog: <https://blog.salesforceairesearch.com/coderl/>

